Evaluating Algorithms for Learning Biological Networks

Allister Bernard and Alexander J. Hartemink
Department of Computer Science
Duke University, Durham, NC 27708

Introduction

In our group we have often encountered the need to evaluate the efficacy of our reverse engineering algorithms. Our evaluation attempts can be divided into two categories: 1. evaluations using simulation studies of synthetically generated data and 2. evaluations using experimentally collected data. Here, we discuss our experiences with both these categories through a set of case studies. The case studies will describe some of the problems we have encountered and lessons we have learned. All these studies involve the reverse engineering of biological networks from data. Most examples are drawn from our experiences with learning regulatory networks, but we also discuss some ongoing work on learning protein-protein interaction networks. With respect to regulatory networks, we discuss the learning of dynamic and static regulatory networks from synthetic and experimental data. The two biological networks we examine are the cell cycle in yeast and the vocal communication system in the songbird brain. For both these examples we used graphical models, so our evaluation studies will be focused on the learning of such networks using graphical models. With respect to protein-protein interaction networks, we discuss some of the difficulties we have encountered in comparing our work with other algorithms that have been published in the literature.

Evaluations using Synthetic Data

Simulations for Dynamic Regulatory Networks

We were interested in studying the vocal communication system of songbirds because they have the rare trait of vocal learning [4, 3, 5]. As experimental data was not yet available, we decided to test our methods with simulation studies where the underlying truth was known in advance. This was done in part to help us determine the best way to collect experimental data. We simulated noisy electrophysiological activity data and noisy gene expression data for several regions of the brain. The goal of this exercise was to learn regulatory networks for different regions of the brain as the bird exhibits a behavior such as singing. This data is dynamic as birds exhibit behavioral changes over time. We then reverse engineered these regulatory networks using algorithms for learning dynamic Bayesian networks from data. We evaluated the robustness of this method to numerous choices of settings including the true underlying network topology, the quantity of data, the sampling interval for observing the data, model scoring metrics, and the discretization method applied to the data, as shown in figure 1. We briefly highlight some lessons we learned below.

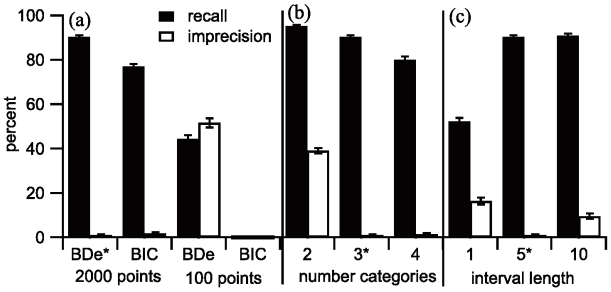

Figure 1: Evaluating **(a)** scoring metrics, **(b)** discretization, and **(c)** sampling intervals. Asterisks indicate the basic configuration (BDe scoring metric, 2000 data points, 3-category discretization and sampling interval of 5) varied along individual dimensions to produce the three panels (a), (b), and (c).

L1 Our simulation studies were not only useful in

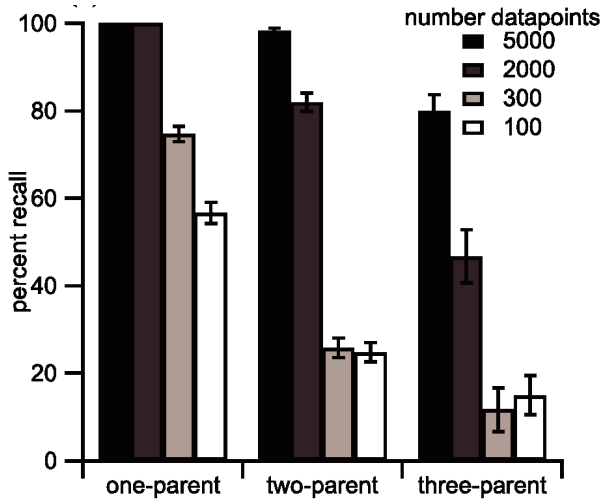

Figure 2: Effect of quantity of data and the number of parents on percentage recall.

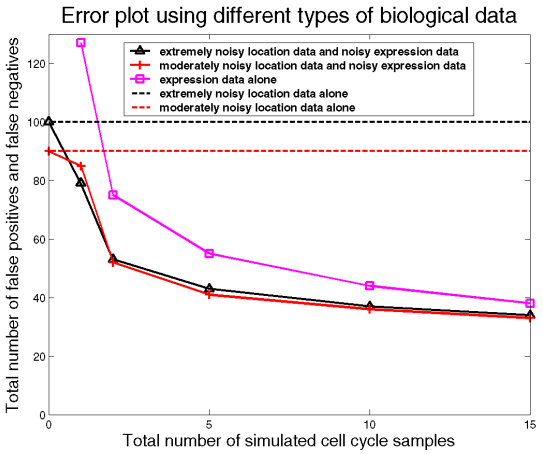

Figure 3: Total number of errors while learning a synthetic cell cycle using (noisy simulated) expression and location data, separately and with both types of data together. The graph shows the effect of increasing the number of cell cycles worth of expression data, both with and without location data.

evaluating our algorithms, but also in guiding the experimental data collection process.

L2 Reverse engineering of networks when genes are regulated by one parent is easier than that when genes are regulated by multiple parents, as shown in figure 2.

L3 Reverse engineering of common biological structures like feedback loops, cascades, and multiple targets does not appear to pose a significant obstacle for the learning procedure.

L4 We observed that there exists an optimal sampling interval for reverse engineering networks. This optimal interval appears to be related to the natural dynamics of the system by which we mean the average time required for one node to propagate its influence to another node directly. A sampling interval that is too long adversely effects recall and a sampling interval that is too short can adversely effect imprecision. We found that simple linear interpolation can reduce imprecision when learning from small quantities of data. When the sampling interval was longer that the natural dynamics of the system we encountered a phenomenon we called 'link-jumping' where we learned a single (indirect) link that actually represented multiple (direct) links.

Simulations for Joint Learning of Regulatory Networks

In addition to the simulation study for songbirds, we conducted a similar study for the yeast cell cycle. We conducted this study to evaluate the efficacy of our joint learning algorithm for learning dynamic regulatory networks. Our algorithm learned these networks jointly from gene expression and transcription factor binding (location) data [1]. We developed an extension to the scoring metric for learning dynamic Bayesian networks by using an informative prior to incorporate the location data. We then generated simulated expression and location data having different noise characteristics. We studied the ability of our algorithm to reverse engineer the underlying simulated regulatory network. This simulation study taught us two main lessons.

L5 Joint learning from multiple types of data was better at reverse engineering the underlying network than methods using each type of data alone, as shown in figure 3.

L6 Our joint learning method was able to reverse engineer networks using less data. To reverse engineer a network with the same number of

errors with only one type of data would require more data, as seen in figure 3.

Evaluations using Experimental Data

A 'Bronze Standard' for the Yeast Cell Cycle

In the same study on joint learning, in addition to studying simulated data, we also tried to evaluate our algorithm using experimentally collected yeast data [1]. In order to do this, we selected a set of representative proteins known to be regulators or regulated during the cell cycle. We then queried various yeast databases to construct a true underlying regulatory network. This network is obviously not the complete underlying truth for the representative set of proteins but served as a 'bronze standard' representing the partial true underlying network. This bronze standard is not the absolute truth but could contain errors in the form of both exclusions and/or inclusions.

L7 As in **L5**, we were able to show that our joint learning algorithm performed better than algorithms that would learn networks using each type of data alone. This was done by comparing the total number of true positives, true negatives, false positives, and false negatives recovered by our method to those using only expression or location data. This comparison was made with respect to the 'bronze standard' and the results are shown in table 1.

L8 Our joint learning approach served as an example of a method which integrated protein-DNA interaction networks with transcriptional networks. Here, nodes have the dual interpretation of proteins and genes while edges have the dual interpretation of a protein-DNA interaction or a regulatory role.

Table 1: Comparison of the highest scoring networks found with the 'bronze standard' network.

Experiment	**TP**	**TN**	**FP**	**FN**
Expression only	7	181	20	32
Location only	25	184	17	14
Expression + Location	28	189	12	11

Prediction of Protein-Protein Interactions

A final example is the prediction of protein-protein interaction networks from various types of experimental data [2]. This work is research in progress and so we do not have final results to discuss. However we have encountered some obstacles in evaluating our methodology and will enumerate those here. In an attempt to compare our work to published work in the literature, we discovered that a number of published results were themselves not directly comparable. This happened because they usually differed from each other along one of two dimensions, namely differences in training sets and differences in test sets. For training sets, some methods use examples of non-interactions to improve an algorithm's predictive power. Thus comparing methods that use non-interactions with those that do not would not be fair. More importantly, there is no well-defined procedure for selecting a set of non-interactions. For test sets, we noticed that the types of interactions/non-interactions used could vary dramatically from one published method to the next. Such problems suggest a need to develop standardized training and test sets to help the community compare and evaluate their methods.

Discussion

Simulation studies have proved to be useful in providing insight into the strengths and weaknesses of reverse engineering algorithms. This can help not only in designing better algorithms, but also in experimental data collection protocols. However simulations could also provide a misleading picture as it is difficult to realistically simulate the underlying true process. On the other hand, increasing the complexity of the simulation could also be unrealistic and lead to an unnecessary amount of time wasted on an unfruitful venture.

The biggest problem with evaluations using experimental data for learning biological networks is that not much is known about the underlying truth. Thus it is hard to assess the efficacy of an algorithm. Partial knowledge can hide the abilities of good and

bad methods alike. It then becomes difficult to compare different methods. In the face of partial truth, evaluations need to be done on a representative subset of the network, which leads to the natural question of what subset should be selected. This subset of the network is usually dependent on the learning problem being studied. Selecting a general representative subset for all problems can result in similar drawbacks in our evaluations as in the case of simulation studies. Even if a 'bronze standard' is eventually agreed upon, what happens when we need to update the 'bronze standard' in the face of new information, as will inevitably happen?

Simulations studies are always useful but evaluations using experimental data are critical. To improve this evaluation process we should develop standardized test sets wherever possible. These can then be used to evaluate the efficacy of any reverse engineering algorithm. Additionally there should be a standardized way to report results. This will not only aid the process of comparing two methods but also facilitate our understanding of the performance of a single method as the standardized test sets change over time. Based on our experience, we feel that a good candidate for introducing gold standards would be the model organism *S. cerevisiae*. Yeast would be an ideal candidate as it has been heavily studied. Additionally a wide range of experimental data have been collected and made available for this organism.

In our joint learning algorithm, we examined methods for integrating networks in which nodes and/or edges can mean different things. A possible step further would be to look at the development of visualization tools to aid the presentation of such networks.

Developing visualization tools as well as gold standards, to aid the reverse engineering designer is a hard task due to the variety of biological problems studied along with the numerous types of information available. We believe a first step in this direction is better than no step at all.

References

[1] Allister Bernard and Alexander J. Hartemink. Informative structure priors: Joint learning of dynamic regulatory networks from multiple types of data. In *PSB*, pages 459–470, 2005.

[2] Allister Bernard, David Vaughn, and Alexander J. Hartemink. Modeling protein-protein interaction assays. in preparation, 2006.

[3] V. Anne Smith, Alexander J. Hartemink, and Erich Jarvis. Influence of topology and data collection on functional network inference. *PSB*, 8:164–175, 2003.

[4] V. Anne Smith, Erich Jarvis, and Alexander J. Hartemink. Evaluating functional network inference using simulations of complex biological systems. *ISMB*, 18:S216–S224, 2002.

[5] Jing Yu, V. Anne Smith, Paul Wang, Alexander J. Hartemink, and Erich Jarvis. Advances to Bayesian network inference for generating causal networks from observational biological data. *Bioinformatics*, 20:3594–3603, 2004.